Octave - An Open-Source tool (Similar to Matlab)

KR Chowdhary
Former Professor & Head
Email: kr.chowdhary@gmail.com
Webpage:www.krchowdhary.com

Department of Computer Science and Engineering
MBM Engineering College, Jodhpur

May 27, 2015

Simple Commands

To plot more ccurate graph:

```
octave:##> angles=linspace(0, 2*pi, 1000);
octave:##> y = sin(angles);
octave:##> plot(angles, y);
```

Improving the presentation:

```
octave:##> title('Graph of y = sin(x)')
octave:##> xlabel('Angles')
octave:##> ylabel('value')
octave:##> grid on
```

multiple graphs:

```
octave:##> plot(angles, y, ':', angles,
      cos(angles), '-')
```

Simple Commands

To save the graph in a file:

```
octave:##> print('graph1.eps','-deps')
to save postscript version of the graph
```

Graph commands can be saved in file and run later:

```
octave:##> gedit   rectsin.m
t = linspace(0, 10, 100);
y = abs(sin(t));
plot(t,y);

title('Rectified Sine Wave');
xlabel('t');
```

and run it by command

```
octave:##> rectsin<cr>
```

Control statements

```
if expression
    statements
elseif expression
    statements
else
    statements
end

octave:##> a=0; b=2;

octave:##> if a > b
           c=3
      else
           c=4
      end
c = 4

octave:##> 1 == 2
ans=0
```

Control statements

```
switch x
case x1,
    statements
case x2,
   statements
otherwise,
   statements
end

octave:##> a=1;
octave:##> switch a
  case 0
      disp('a is zero');
  case 1
     disp('a is one');
otherwise
     disp('a is not a binary digit');
end

a is one
```

Control statements

```
for variable = vector
    statements
end

octave:##> for n=1:5
   nf(n) = factorial(n);
 end
octave:##> disp(nf)
   1   2     6     24    120

while expression
  statements
end

For example,
octave:##> x=1;
octave:##> while 1+x > 1
    x = x/2;
end
octave:##> x
x = 1.1102e-016
```

Matrices and vectors

```
octave:##> A=[5 7 9
             -1 3 -2]
A = 5 7  9
  -1 3  -2
octave:##> B=[2 0; 0 -1; 1 0]
B= 2    0
   0    -1
   1    0
octave:##> C = [1:3; 8:-2:4]
C = 1  2  3
   8   6  4
Matrix multiplication
octave:##> A*B
ans = 19 -7
     -4  -3
octave:##>B*C
ans = 2  4  6
     -8 -6 -4
      1   2 3
```

The transpose operator

```
octave:##> A
A = 5 7 -1
    3 9 -2
octave:##> A'
ans = 5  3
      7   9
      -1  -2

octave:##> I = eye(4)
I =  1  0  0  0
     0  1  0  0
     0  0  1  0
     0  0  0  1
```

Putting several graphs in one window

```
 subplot(rows ,columns ,select )
 octave:##> x = linspace(-10, 10);
 octave:##> subplot(2,1,1) % Specify two rows, one column, and select
octave:##>  % the top one as the current graph
octave:##> plot(x, sin(x));
octave:##> subplot(2,1,2);
octave:##> plot(x, sin(x)./x);

3D plots:
octave:##> t = 0:pi/50:10*pi;
octave:##> x = sin(t); y = cos(t); z = t;
octave:##> plot3(x, y, z);
```

Putting surfaces

```
octave:##> x = 2:0.2:4; % Define the x- and y- coordinates
octave:##> y = 1:0.2:3; % of the grid lines
octave:##> [X,Y] = meshgrid(x, y); %Make the grid
For example, to plot f(x, y) = (x - 3)^2 - (y - 2)^2 over the grid
calculated earlier, you would type:
octave:##> Z=(X-3).^2 - (Y-2).^2;
octave:##> surf(X,Y,Z)
```